



Iniciación a los motores paso a paso

Usaremos las siguientes maquetas

PAP 1: Uno + ULN2803 + Stepper Unipolar 12V 7,5°

PAP 2: Uno + 2*ULN2803 + 2*Stepper Unipolar 5V reduccion

PAP 3: Duemilanove + A4988 + Nema17 400p 0,9°

PAP 4: Uno + A4988 + Nema17 200p 1,8° + encoder

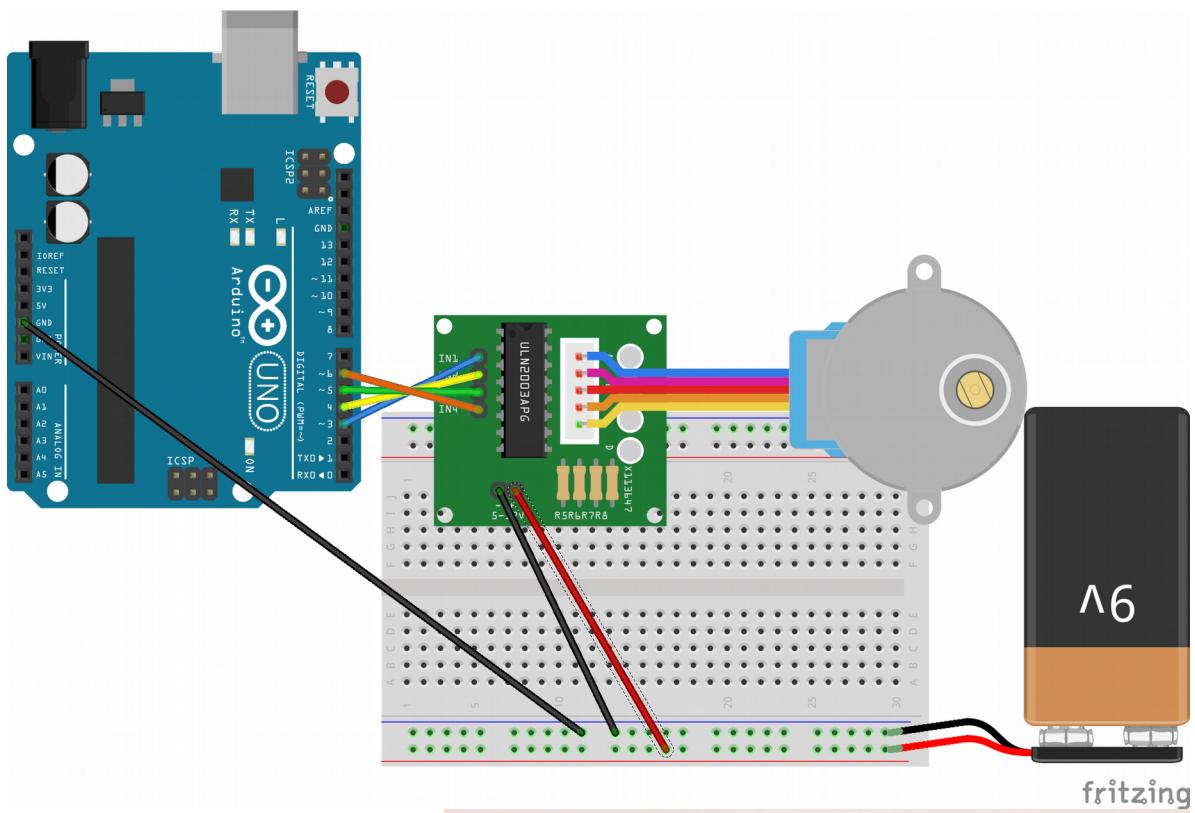
PAP 5: Uno + Cncshield DVD

PAP 6: Mega + RAMPS + Nema 17

PAP 7: Mega + RAMPS + 2 DVD

PAP 1

Uno + ULN2803 + Stepper Unipolar 12V 7,5°



fritzing



```
/*
Arduino Day Zaragoza 2017
Charla Factoria Maker
Iniciación a los motores Paso a Paso
Ejemplo para la maqueta PAP 1
Un motor unipolar de 12V y 48 pasos por vuelta
utiliza la librería Stepper que ya viene en el IDE
arduino
 */

#include <Stepper.h>

const int stepsPerRevolution = 48; // pasos por vuelta

// inicializa la librería stepper con los pines 4-7
Stepper motor(stepsPerRevolution, 4, 6, 5, 7);

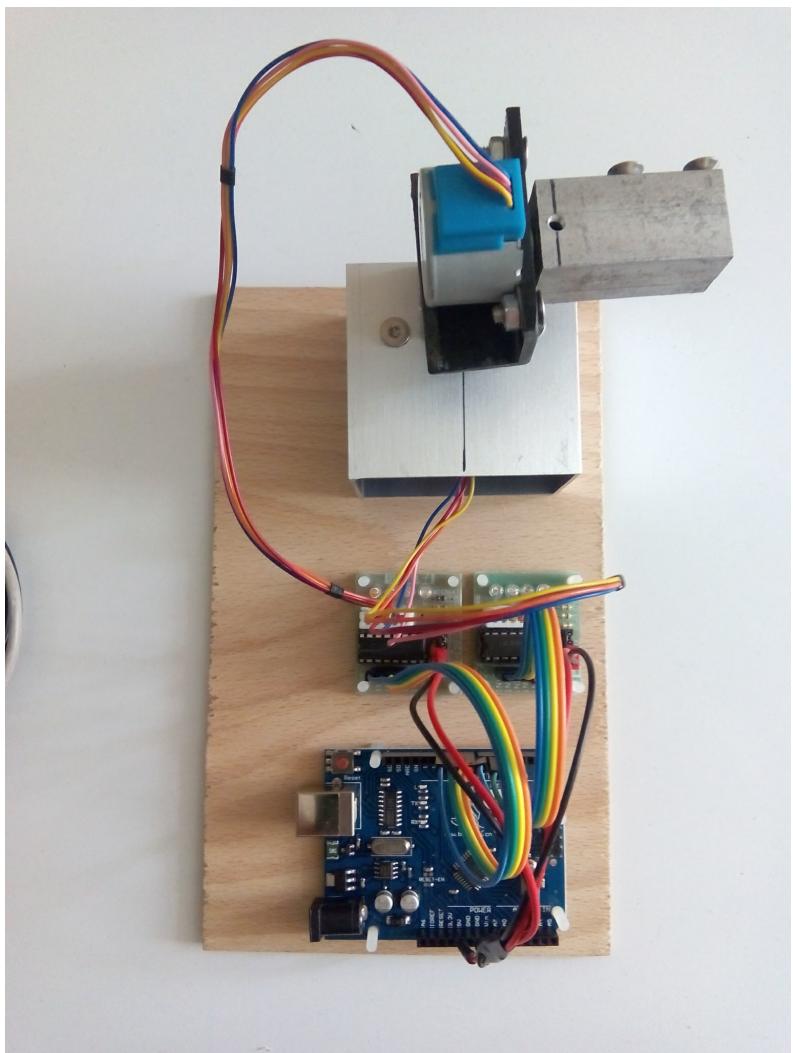
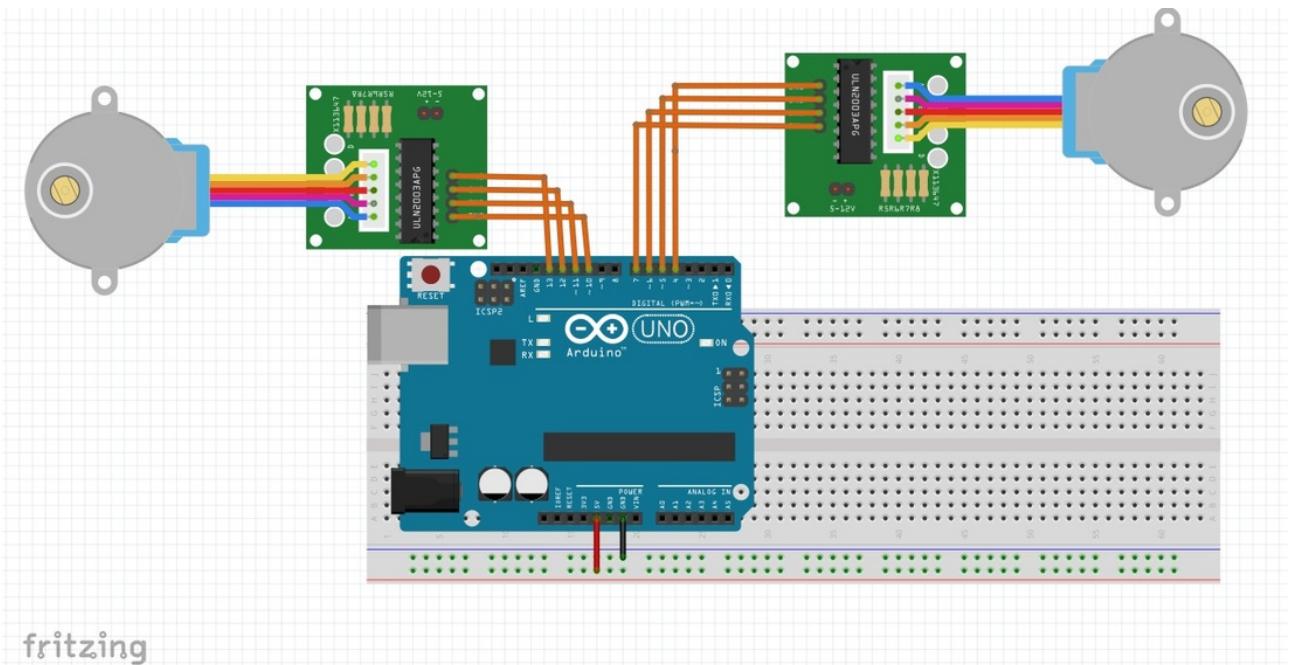
void setup() {
    motor.setSpeed(10);
}

void loop() {
    motor.setSpeed(10);
    motor.step(stepsPerRevolution);
    delay(500);

    motor.setSpeed(3);
    motor.step(-stepsPerRevolution);
    delay(500);
}
```

PAP 2

Uno + 2*ULN2803 + 2*Stepper Unipolar 5V reduccion



```
/*
Arduino Day Zaragoza 2017
Charla Factoria Maker
Iniciación a los motores Paso a Paso
Ejemplo para la maqueta PAP 2
Dos motores unipolares de 5V con reductor, para azimut y
cenit
utiliza la librería Stepper que ya viene en el IDE
arduino
*/
#include <Stepper.h>

const int stepsPerRevolution = 1000;

Stepper cenit(stepsPerRevolution, 8, 10, 9, 11);
Stepper acimut(stepsPerRevolution, 4, 6, 5, 7);

void setup() {
    cenit.setSpeed(20);
    acimut.setSpeed(20);
}

void loop() {
    cenit.step(stepsPerRevolution);
    delay(500);
    acimut.step(stepsPerRevolution);
    delay(500);

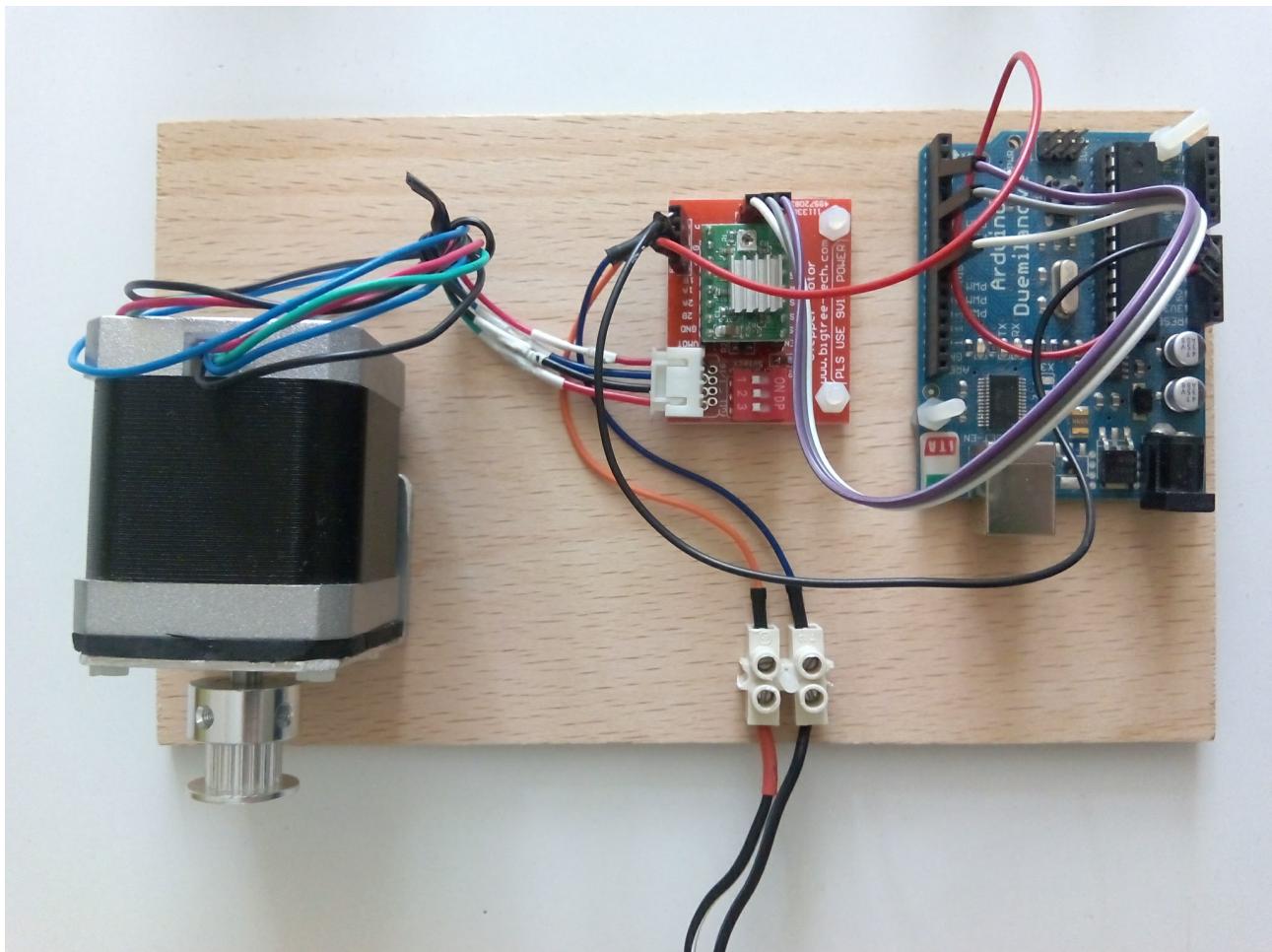
    cenit.step(-stepsPerRevolution);
    delay(500);
    acimut.step(-stepsPerRevolution);
    delay(500);
}
```

PAP 3

Duemilanove + A4988 + Nema17 400p 0,9°



Resolucion		Pines M0, M1 y M2		
A4988	DRV8825	MODE0	MODE1	MODE2
Full step	Full step	Low	Low	Low
1/2 step	1/2 step	High	Low	Low
1/4 step	1/4 step	Low	High	Low
1/8 step	1/8 step	High	High	Low
-	1/16 step	Low	Low	High
-	1/32 step	High	Low	High
-	1/32 step	Low	High	High
1/16 step	1/32 step	High	High	High



```

/*
Arduino Day Zaragoza 2017
Charla Factoria Maker
Iniciación a los motores Paso a Paso
Ejemplo para la maqueta PAP 3
Control de motor bipolar mediante pololu. Placa con interruptores
El pololu necesita tres señales de control: Enable, Dirección y
Paso
Usamos las mismas direcciones que la cncshield con Arduino Uno
El pololu se controla con las señales Dir_Pin_X (pata 5) y
Step_Pin_X (pata 2)
La señal Enable está activa con nivel bajo (LOW) y con HIGH se
desactiva el motor
*/

```

```

int EnPin = 8;           // cncshield enable de los 4 pololus al pin 8
int Dir_Pin_X = 5;      // cncshield dirección del motor X al pin 5
int Step_Pin_X = 2;     // cncshield paso del motor X al pin 2

bool Dir_estado = LOW;   // Para cambiar de dirección
int contador = 0;
void setup() {
    pinMode(EnPin, OUTPUT);          // Define el pin como salida
    pinMode(Dir_Pin_X, OUTPUT);      // Define el pin como salida
    pinMode(Step_Pin_X, OUTPUT);      // Define el pin como salida
    digitalWrite(EnPin, LOW);         // Enable activo con LOW
    digitalWrite(Dir_Pin_X, Dir_estado); // Saca el estado inicial
}

void loop() {              // Paso un 1 y luego un 0
    digitalWrite(Step_Pin_X, HIGH); // Activa Step
    delay(2);                   // Espera un poco
    digitalWrite(Step_Pin_X, LOW); // Desactiva Step
    delay(1);                   // Espera un poco
    contador = contador + 1;
    if (contador == 400) {
        Dir_estado = !Dir_estado; // Invierte la dirección
        contador=0;             // Reinicia el contador
        digitalWrite(Dir_Pin_X, Dir_estado); // Saca Dir
    }
}

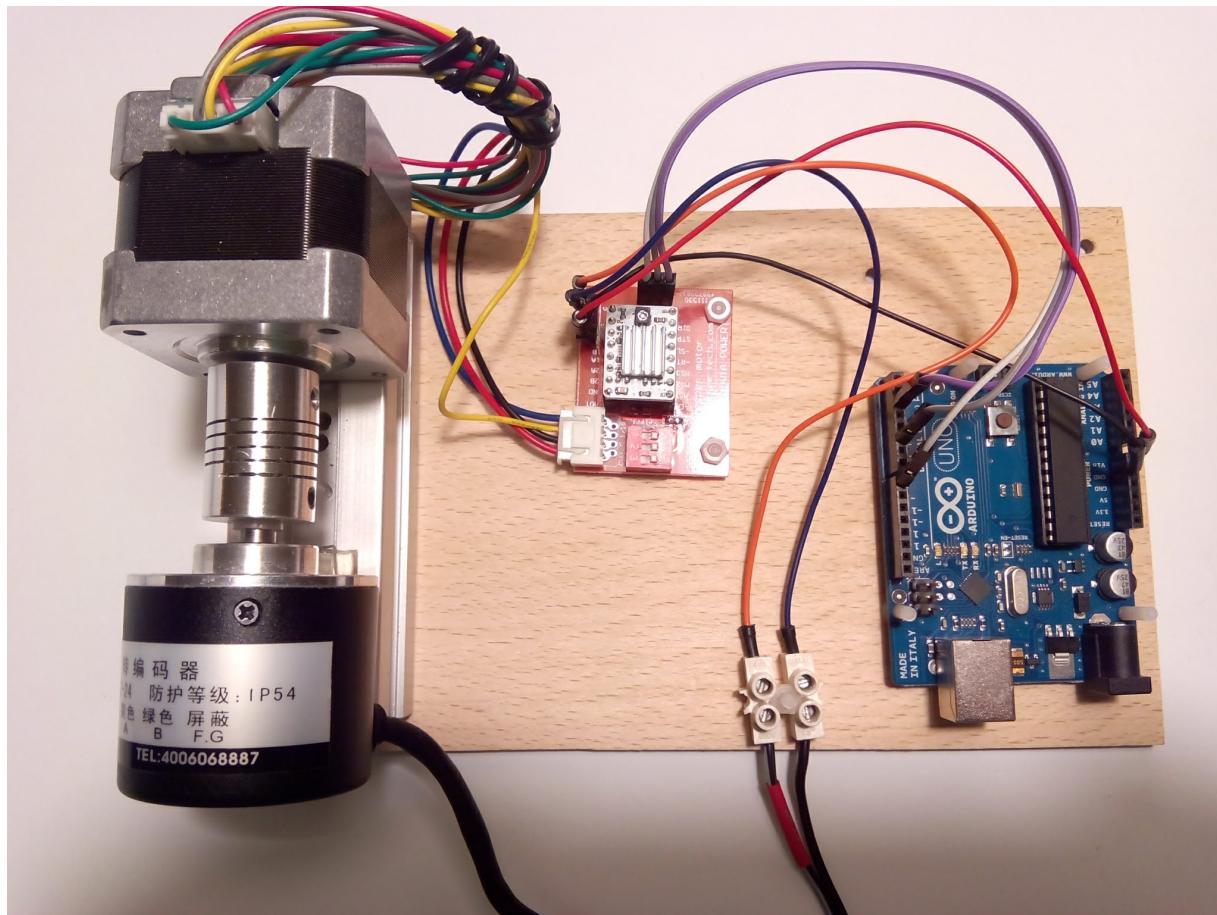
```

PAP 4

Uno + A4988 + Nema17 200p 1,8° + encoder



Resolucion		Pines M0, M1 y M2		
A4988	DRV8825	MODE0	MODE1	MODE2
Full step	Full step	Low	Low	Low
1/2 step	1/2 step	High	Low	Low
1/4 step	1/4 step	Low	High	Low
1/8 step	1/8 step	High	High	Low
-	1/16 step	Low	Low	High
-	1/32 step	High	Low	High
-	1/32 step	Low	High	High
1/16 step	1/32 step	High	High	High



```

/*
Arduino Day Zaragoza 2017
Charla Factoria Maker
Iniciación a los motores Paso a Paso
Ejemplo para la maqueta PAP 4
Control de motor bipolar mediante pololu. Placa con interruptores
El pololu necesita tres señales de control: Enable, Dirección y
Paso
Usamos las mismas direcciones que la cncshield con Arduino Uno
El pololu se controla con las señales Dir_Pin_X (pata 5) y
Step_Pin_X (pata 2)
La señal Enable está activa con nivel bajo (LOW) y con HIGH se
desactiva el motor
*/

```

```

int EnPin = 8;           // cncshield enable de los 4 pololus al pin 8
int Dir_Pin_X = 5;      // cncshield dirección del motor X al pin 5
int Step_Pin_X = 2;     // cncshield paso del motor X al pin 2

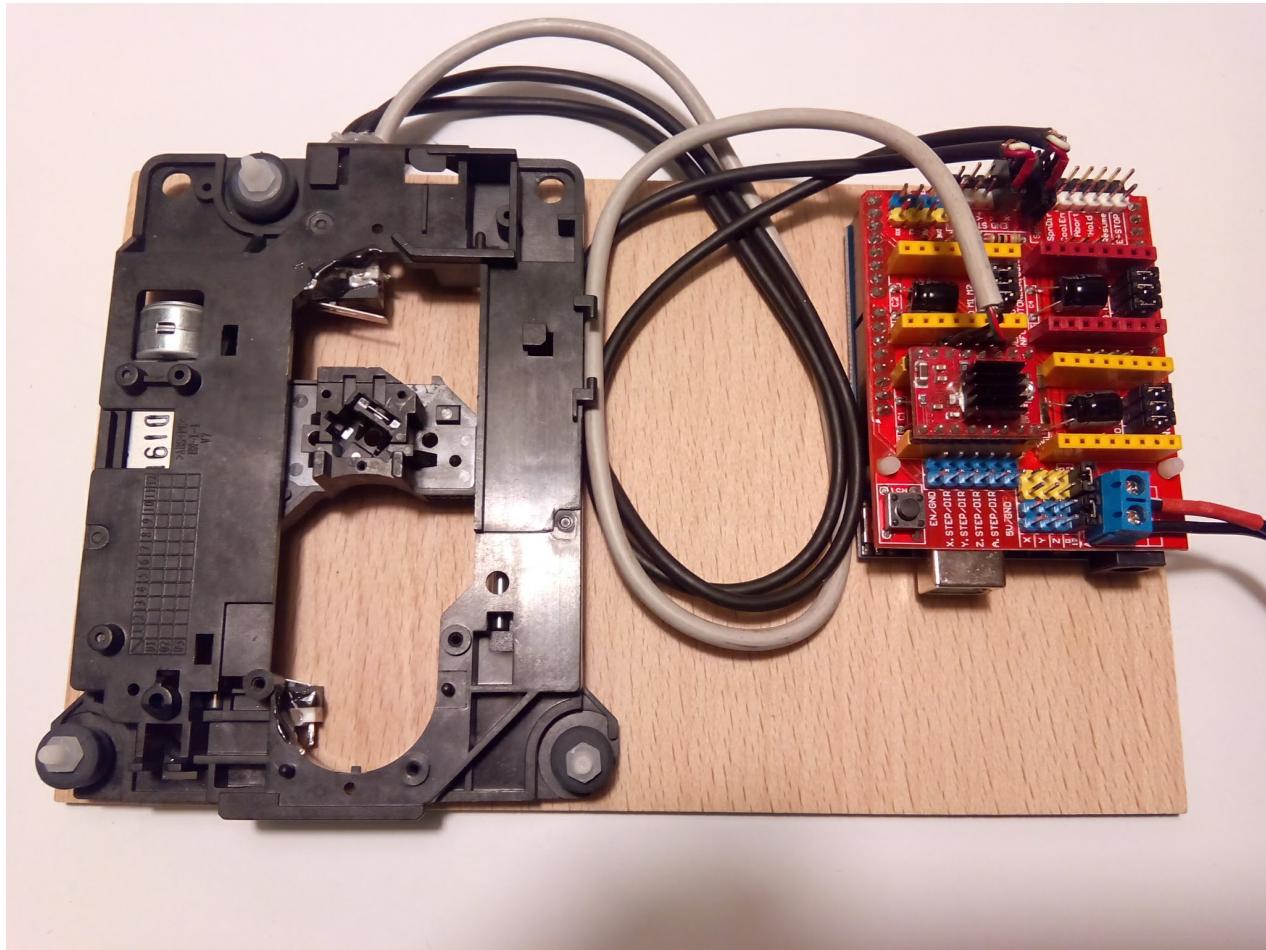
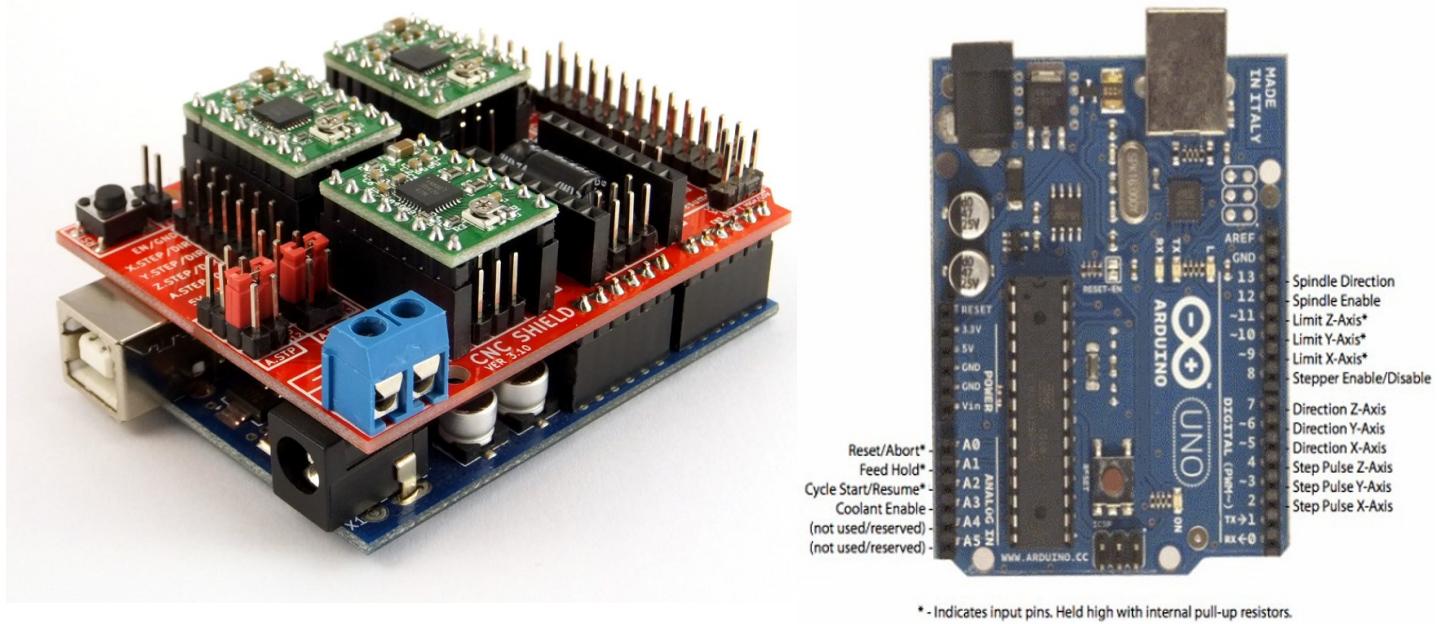
bool Dir_estado = LOW;   // Para cambiar de dirección
int contador = 0;
void setup() {
    pinMode(EnPin, OUTPUT);          // Define el pin como salida
    pinMode(Dir_Pin_X, OUTPUT);      // Define el pin como salida
    pinMode(Step_Pin_X, OUTPUT);      // Define el pin como salida
    digitalWrite(EnPin, LOW);         // Enable activo con LOW
    digitalWrite(Dir_Pin_X, Dir_estado); // Saca el estado inicial
}

void loop() {              // Paso un 1 y luego un 0
    digitalWrite(Step_Pin_X, HIGH); // Activa Step
    delay(2);                   // Espera un poco
    digitalWrite(Step_Pin_X, LOW); // Desactiva Step
    delay(1);                   // Espera un poco
    contador = contador + 1;
    if (contador == 400) {
        Dir_estado = !Dir_estado; // Invierte la dirección
        contador=0;             // Reinicia el contador
        digitalWrite(Dir_Pin_X, Dir_estado); // Saca Dir
    }
}

```

PAP 5

Uno + Cncshield DVD



```

/*
Arduino Day Zaragoza 2017
Charla Factoria Maker
Iniciación a los motores Paso a Paso
Ejemplo para la maqueta PAP 5
Control de motor bipolar mediante Arduino Uno y cncshield.
El pololu necesita tres señales de control: Enable, Dirección y
Paso
El pololu se controla con las señales Dir_Pin_X (pata 5) y
Step_Pin_X (pata 2)
La señal Enable está activa con nivel bajo (LOW) y con HIGH se
desactiva el motor
Este programa genera una serie de pulsos en la salida Step_Pin_X
(Hay que ponerlo a 1, esperar un tiempo, ponerlo a 0)
Cuando ha mandado un número de pulsos, cambia la dirección
No usa los finales de carrera
*/

```

```

int EnPin = 8;           // enable de los 4 pololus al pin 8
int Dir_Pin_X = 5;      // señal de dirección del motor X al pin 5
int Step_Pin_X = 2;      // señal de paso del motor X al pin 2

bool Dir_estado = LOW;   // Para cambiar de dirección
int contador = 0;
void setup() {
    pinMode(EnPin, OUTPUT);          // Define el pin como salida
    pinMode(Dir_Pin_X, OUTPUT);      // Define el pin como salida
    pinMode(Step_Pin_X, OUTPUT);      // Define el pin como salida

    digitalWrite(EnPin, LOW);         // El pololu activo con LOW
    digitalWrite(Dir_Pin_X, Dir_estado); // Saca el estado inicial
}

void loop() {               // Paso: mandar un 1 y luego un 0
    digitalWrite(Step_Pin_X, HIGH); // Activa el pin de Paso
    delay(1);                   // Espera un poco
    digitalWrite(Step_Pin_X, LOW); // Desactiva el pin de Paso
    delay(1);                   // Espera un poco
    contador = contador + 1;
    if (contador == 2400) {
        Dir_estado = !Dir_estado; // Invierte la dirección
        contador=0;              // Reinicia el contador
        digitalWrite(Dir_Pin_X, Dir_estado); // Saca Dir
    }
}

```

```

/*
Arduino Day Zaragoza 2017
Charla Factoria Maker
Iniciación a los motores Paso a Paso
Ejemplo para la maqueta PAP 5
Control de motor bipolar mediante Arduino Uno y cncshield.
El pololu necesita tres señales: Enable, Dirección y Paso
El pololu se controla con las señales Dir_Pin_X (pata 5) y
Step_Pin_X (pata 2)
La señal Enable está activa con nivel bajo (LOW) y con HIGH se
desactiva el motor
Este programa genera una serie de pulsos en la salida Step_Pin_X
(Hay que ponerlo a 1, esperar un tiempo, ponerlo a 0)
Cuando ha mandado un número de pulsos, cambia la dirección
Los finales de carrera usan un interruptor NO entre el pin y GND
y usan la resistencia de pull-up interna.
Se pueden comprobar con el ejemplo Digital>DigitalInputPullUp
Conectamos los dos finales de carrera en limitX y LimitY para
facilitar el programa.
Comienza con dirección LOW, hacia el final de carrera Y
Cuando toca el final de carrera (limitY=LOW) cambia la dirección
a HIGH y el motor se mueve hacia el final de carrera X
Cuando llega al final de carrera (limitX=LOW) cambia la
dirección a LOW y vuelve a empezar el ciclo
*/

```

```

int retardo = 1; // Duración del pulso
int EnPin = 8; // Señal de enable de los 4 pololus al pin 8
int Dir_Pin_X = 5; // Señal de dirección del motor X al pin 5
int Step_Pin_X = 2; // Señal de paso del motor X al pin 2
int limitX = 9; //Final de carrera
int limitY = 10;
bool Dir_estado = LOW; // Para cambiar de dirección
int contador = 0; // cuenta los pasos
void setup() {
    Serial.begin(9600);
    pinMode(EnPin, OUTPUT); // Define el pin como salida
    pinMode(Dir_Pin_X, OUTPUT); // Define el pin como salida
    pinMode(Step_Pin_X, OUTPUT); // Define el pin como salida
    pinMode(limitX, INPUT_PULLUP); // Activa pull-up interna
    pinMode(limitY, INPUT_PULLUP); // Activa pull-up interna
    digitalWrite(EnPin, LOW); // Activa con nivel bajo
    digitalWrite(Dir_Pin_X, Dir_estado); // Saca Dir
}

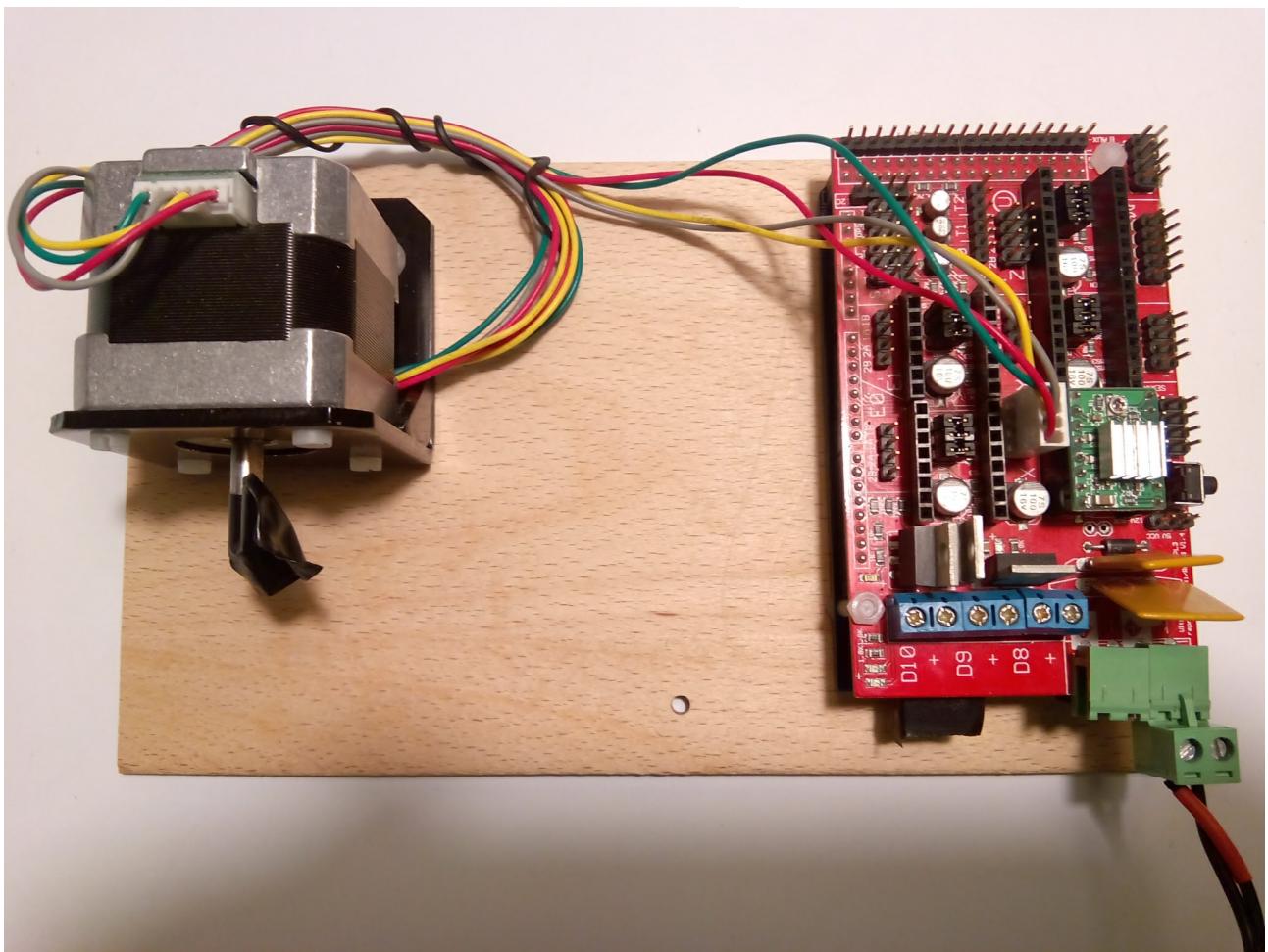
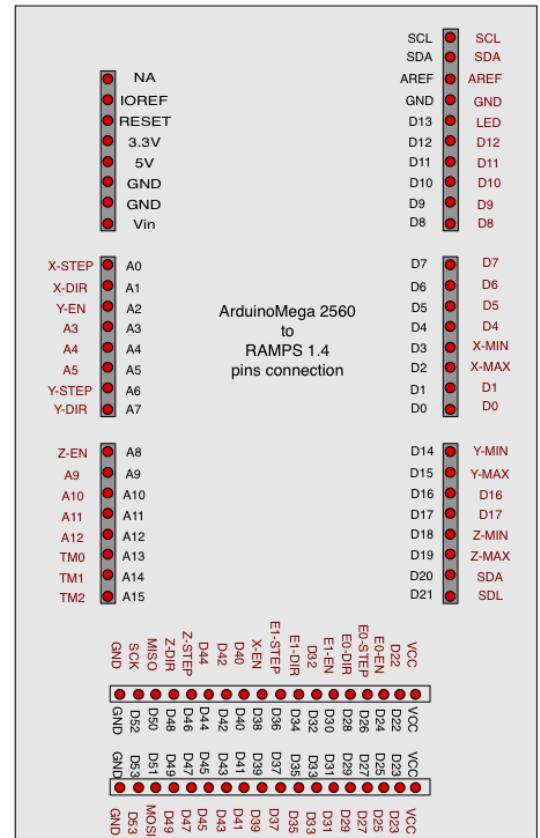
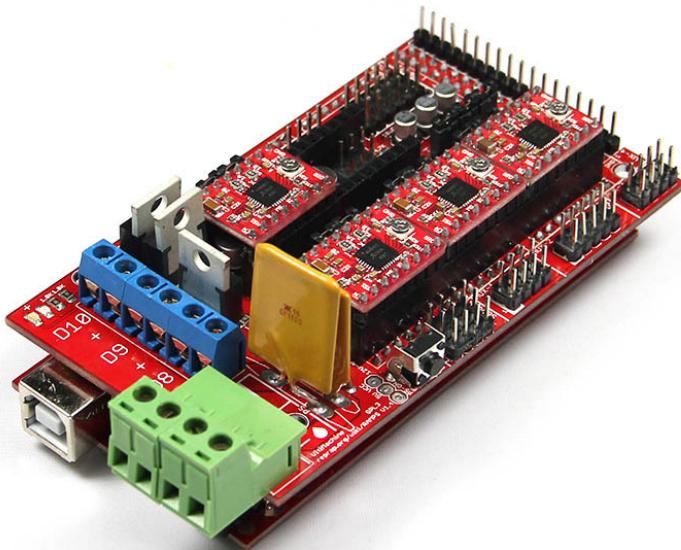
void loop() { // Paso: un 1 y luego un 0
    int sensorValY = digitalRead(limitY); // Lee final de carrera
    int sensorValX = digitalRead(limitX);
    digitalWrite(Step_Pin_X, HIGH); // Activa el pin de Paso
    delay(retardo); // Espera un poco
    digitalWrite(Step_Pin_X, LOW); // Desactiva el pin de Paso
    delay(retardo); // Espera un poco
    contador = contador + 1;
}

```

```
// Si lee LOW es porque el final de carrera está pulsado
if (sensorValX == LOW) {
    if (contador > 10) Serial.println(contador);
    Dir_estado = LOW;
    contador = 0;
}
// Si lee LOW es porque el final de carrera está pulsado
if (sensorValY == LOW) {
    if (contador > 10) Serial.println(contador);
    Dir_estado = HIGH;
    contador = 0;
}
digitalWrite(Dir_Pin_X, Dir_estado);      // Saca Dir
}
```

PAP 6

Mega + RAMPS + Nema 17



```
/*
Arduino Day Zaragoza 2017
Charla Factoria Maker
Iniciación a los motores Paso a Paso
Ejemplo para la maqueta PAP 6
Control de motor bipolar mediante Arduino Mega y RAMPS.
Cada pololu necesita tres señales: Enable, Dirección y Paso
La señal Enable está activa con nivel bajo (LOW) y con HIGH se
desactiva el motor
Para generar los pulsos usamos la librería AccelStepper
Cuando ha girado los pasos indicados le decimos que gire -pasos
*/
```

```
#include <AccelStepper.h>

// The X Stepper pins
// for the Arduino Mega + RAMPS 1.4
#define MOTOR_X_ENABLE_PIN 38
#define MOTOR_X_STEP_PIN 54
#define MOTOR_X_DIR_PIN 55

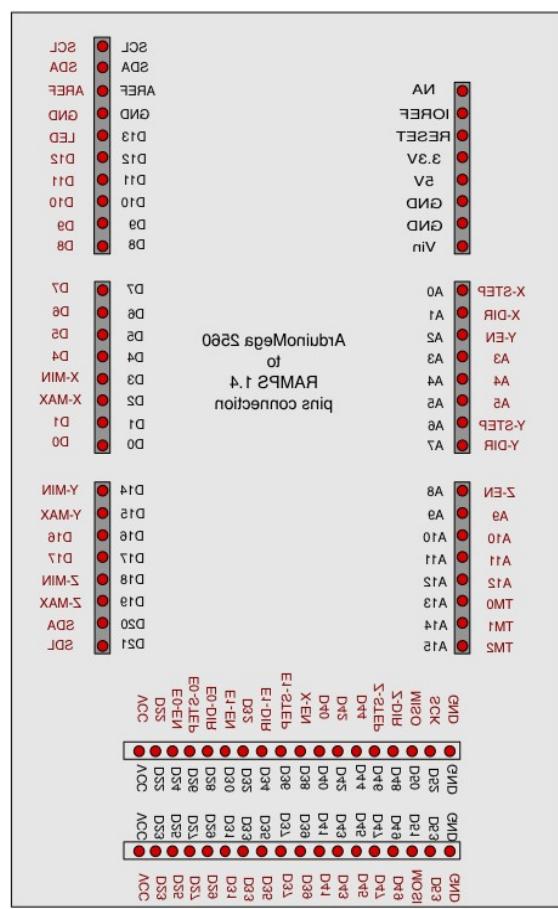
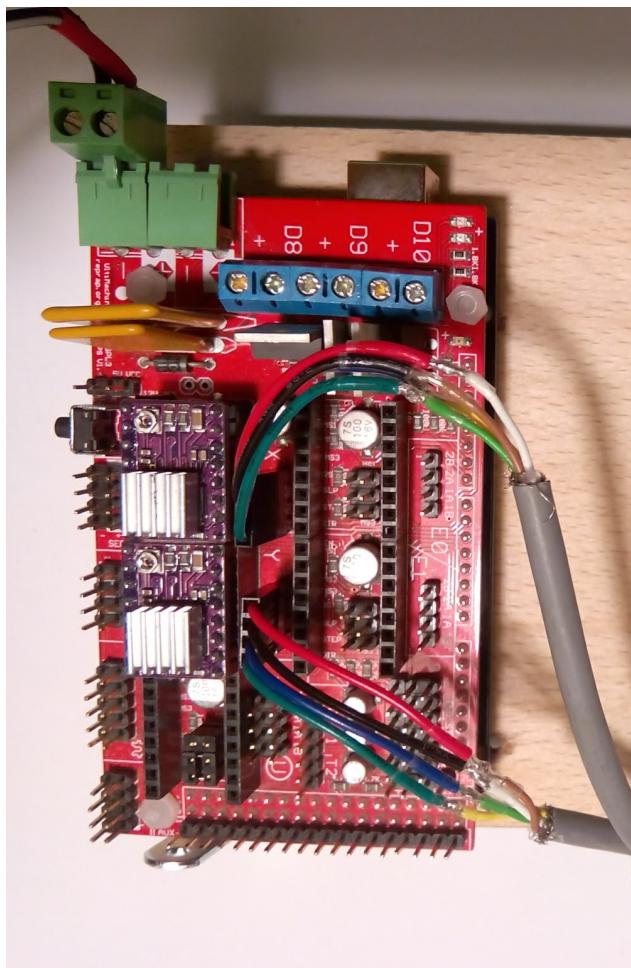
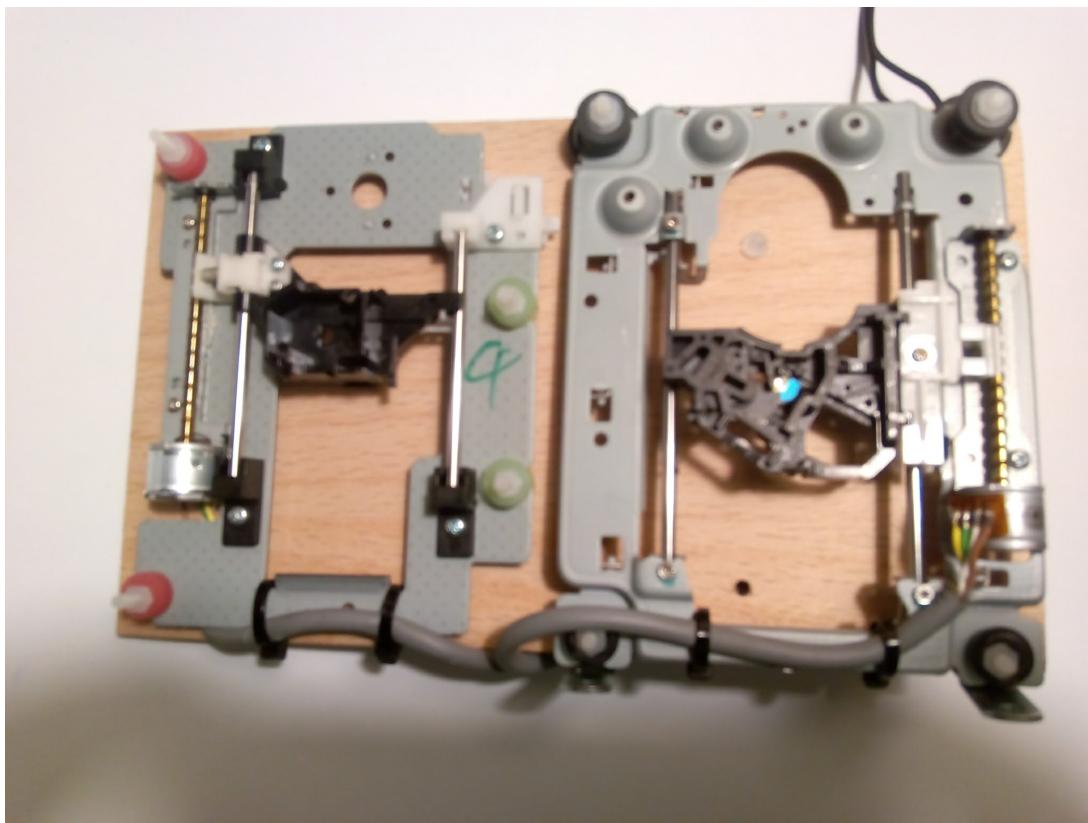
// Define a stepper and the pins it will use
// AccelStepper stepper;
// Defaults to AccelStepper::FULL4WIRE (4 pins) on 2, 3, 4, 5
AccelStepper motorA(1, MOTOR_X_STEP_PIN, MOTOR_X_DIR_PIN);

void setup()
{
    motorA.setEnablePin(MOTOR_X_ENABLE_PIN);
    motorA.setPinsInverted(false, false, true);
    motorA.setAcceleration(10000);
    motorA.move(6400);
    motorA.setMaxSpeed(5000);
    motorA.enableOutputs();
}

void loop()
{
    // If at the end of travel go to the other end
    if (motorA.distanceToGo() == 0)
        motorA.moveTo(-motorA.currentPosition());
    motorA.run();
}
```

PAP 7

Mega + RAMPS + 2 DVD



```

/*
Arduino Day Zaragoza 2017
Charla Factoria Maker
Iniciación a los motores Paso a Paso
Ejemplo para la maqueta PAP 7
Control de 2 motores de DVD mediante Arduino Mega y RAMPS.
Cada pololu necesita tres señales: Enable, Dirección y Paso
Enable activa con nivel bajo. Con HIGH se desactiva el motor
Para generar los pulsos usamos la librería AccelStepper
Cuando ha girado los pasos indicados le decimos que gire -pasos
*/
#include <AccelStepper.h>

// for the Arduino Mega + RAMPS 1.4
#define MOTOR_X_ENABLE_PIN 38
#define MOTOR_X_STEP_PIN 54
#define MOTOR_X_DIR_PIN 55

#define MOTOR_Y_ENABLE_PIN A2
#define MOTOR_Y_STEP_PIN A6
#define MOTOR_Y_DIR_PIN A7

AccelStepper motorX(1, MOTOR_X_STEP_PIN, MOTOR_X_DIR_PIN);
AccelStepper motorY(1, MOTOR_Y_STEP_PIN, MOTOR_Y_DIR_PIN);

void setup()
{
    motorX.setEnablePin(MOTOR_X_ENABLE_PIN);
    motorX.setPinsInverted(false, false, true);
    motorX.setAcceleration(2000);
    motorX.move(3200);
    motorX.setMaxSpeed(10000);
    motorX.enableOutputs();

    motorY.setEnablePin(MOTOR_Y_ENABLE_PIN);
    motorY.setPinsInverted(false, false, true);
    motorY.setAcceleration(30000);
    motorY.move(3200);
    motorY.setMaxSpeed(30000);
    motorY.enableOutputs();
}

void loop()
{
    // Cuando el motor X llega a un extremo, invierte el sentido
    if (motorX.distanceToGo() == 0)
        motorX.moveTo(-motorX.currentPosition());
    motorX.run();
    // Cuando el motor Y llega a un extremo, invierte el sentido
    if (motorY.distanceToGo() == 0)
        motorY.moveTo(-motorY.currentPosition());
    motorY.run();
}

```